2008 IFIP International Conference on Network and Parallel Computing

Experimental Tests on SCTP over IPSec

Maria-Dolores Cano, Juan A. Romero, Fernando Cerdan
Department of Information Technologies & Communications
Technical University of Cartagena (UPCT)
Cartagena, Spain
{mdolores.cano, juan.romero, fernando.cerdan}@upct.es

Abstract

As telecommunication technologies evolve, security in communications becomes a more and more relevant issue. IPSec is a set of protocols aiming to enhance security at the IP layer. Specifically, IPSec and IKE are important security mechanism that provide cryptographic-based protection for IP packets, and consequently for IP services. SCTP is a standardized transport protocol whose main features include multihoming and multistreaming, and is gaining momentum as a general-purpose transport protocol. While the simultaneous use of these two protocols is feasible, it is under study how to make them work efficiently. In this paper, we present a simple method to improve SCTP-IPSec-IKE compatibility by modifying the structure of the Security Associations. Despite the conceptual simplicity of our proposal, it has not been proposed before in related literature.

1. Introduction

Security concerns are still a hot topic in current communications network research. On the one hand, IPSec (Internet Protocol Security) [1] is a security protocol at the IP layer aiming to protect upper layers information and IP header fields from unauthorized access. Before the IPSec secure communication is established, both entities should agree on what security parameters and protocols will be used. IKE (Internet Key Exchange) [2] offers a secure and transparent method to carry out this negotiation, and it is usually employed together with IPSec. On the other hand, SCTP (Stream Control Transmission Protocol) [3] is a transport protocol no longer exclusive for telephone signaling, but it is becoming more and more used as a general purpose data transport protocol. In heterogeneous network environments, it is usual to suggest the use of IPSec to protect SCTP communications [4], and so does in wireless communications using mobile IP [5].

The combined use of these elements, IKE, IPSec, and SCTP, implies some drawbacks. Briefly, IPSec was not designed to be compatible with multihoming. To tackle this problem, several approaches have been proposed. For instance, security could be provisioned by means of TLS (Transport Layer Security), but the delay introduced in SCTP/TLS communications is much longer than with SCTP/IPSec as demonstrated in [6]. Another idea is to modify the SCTP protocol itself to incorporate security, as proposed in [7] [8]. Proposal from [7] may suffer from a high loss probability in wireless networks. In [8], authors introduced an enhanced version of [7], using a collaborative approach to improve the transmission performance, and obtaining good results. However, IPSec can provide security at the network layer without the need to change the SCTP protocol. The RFC3554 [9] shows the specifications for an IPSec implementation able to work properly with SCTP. Nevertheless, from our point of view, the implementation detailed in [9] can be improved in order to really benefit from all SCTP characteristics, especially from multihoming.

In this paper we propose an enhanced IPSec-IKE implementation partially based on the theoretical specifications proposed in [9], designed to work with SCTP in a transparent way. This enhancement allows two SCTP endpoints to use any IP address within a pool of IP addresses (e.g. IP addresses within the same network) keeping the same IPSec security terms. During the IKE negotiation phase, each endpoint will communicate its available IP addresses (single ones or from a pool). The main advantage lies on the fact that endpoints can use different IP addresses during the communication (multihoming) but IPSec requirements do not need to be set for every IP address in use.

The rest of this paper is organized as follows. Section II describes the main features of IKE, IPSec,

978-0-7695-3354-4/08 $25.00 © 2008 IEEE
DOI 10.1109/NPC.2008.92

and SCTP, remarking compatibility issues. Next, Section III presents the proposed IPSec and IKE implementations, and shows the experimental tests carried out. Finally, we include the most relevant conclusions derived from this study in Section IV.

2. Protocols overview

In this section we present a brief introduction to the main characteristics of IKE, IPSec, and SCTP, as well as potential compatibility drawbacks.

2.1. IKE

One of the main characteristics of IPSec is that communication entities can negotiate security protocols, ciphering algorithms, and cryptographic keys to define a Security Association (SA), and the current standard to carry out this negotiation is IKE. A Security Association is a unidirectional relationship between a sender and a receiver, which offers specific security services to all traffic belonging to this relation. An IPSec Security Association is defined by the tuple {SPI, destination address, security protocol}. The SPI is a unique local index required if different SAs exist per destination address. The security protocol may be either AH or ESP as we will seen later.

The negotiation protocol IKE is a hybrid that takes characteristics from the key exchange protocols Oakley [10] and SKEME (Secure Key Exchange Mechanism for Internet) [11], and operates within the framework provided by the ISAKMP (Internet Security Association and Key Management Protocol) [12]. IKE is based on a two-phase negotiation. During the first phase, parameters needed to set an ISAKMP Security Association are negotiated. Six messages are exchanged in this phase: first two messages to chose the parameters of the ISAKMP SA, next two messages to exchange keying material (including nonces to avoid replay attacks), and last two messages for authentication. There is another option available for the first phase, called aggressive mode. However, it does not include identity protection, thus it was not selected for this work. During the second phase, the agreed upon ISAKMP SA is used to secure next exchange between entities. In this second phase, entities negotiate security parameters for an IPSec Security Association. To do so, three messages are used. The first message includes an IPSec SA proposal (IPSec protocol, SPI, and security algorithms). With the second message, the receiver accepts the IPSec SA proposal. The third message confirms that the IPSec SA is in use and will be employed to secure all IP data exchange from now on. Fig. 1 represents the complete process.

Each ISAKMP message has a fixed header format followed by one or more payloads (see Fig. 2). Defined payloads are: Security Association, Proposal, Transform, Key Exchange, Identification, Certificate, Certificate Request, Nonce, etc. Fig. 3 shows a generic payload header. For instance, the Identity payload (Fig. 4) contains DOI-specific data used to exchange identification information. This information is used for determining the identities of communicating peers, and may be used for determining authenticity of information. We will use this payload, as we will show later, in order to communicate the peer entity available IP addresses for multihoming.

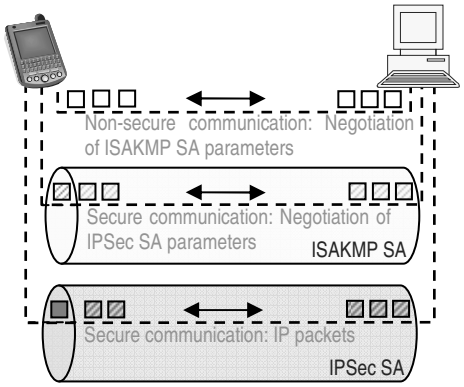

Figure 1. Process of establishing an IPSec Security Association.

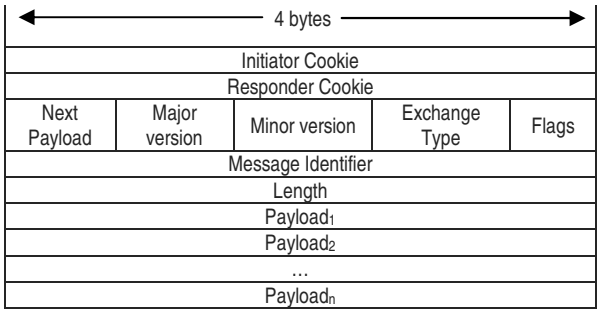

Figure 2. ISAKMP header.

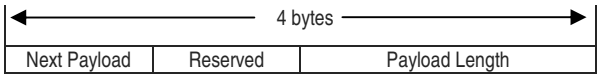

Figure 3. Generic payload header.

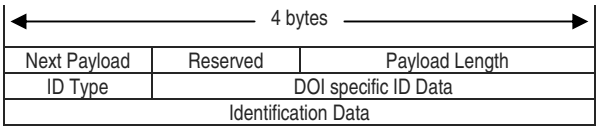

Figure 4. Identity payload. ID Type spedifies the type of identification being used. DOI specific ID Data contains DOI specific Identification data. Identification Data (variable length) contains identity information.

2.2. IPSec

IPSec comprises three protocols: two security protocols (AH, Authentication Header and ESP, Encapsulation Security Payload), and a third protocol (IKE, Internet Key Exchange) aimed to safely exchange security communication parameters between endpoints. AH provides authentication, ESP provides encryption and optionally authentication, and IKE is an implementation of ISAKMP methodology modified to provide a higher security level. Security parameters of an IPSec one-way communication are stored in a SA. Hence, a communication between two endpoints requires two SA at each endpoint. Every time a packet needs to be processed, IPSec is addressed to the SA in order to check security requirements and parameters. All SA are in turn stored in a SA Database (SAD).

2.3. SCTP

SCTP, initially designed for telephone signaling, has become a general purpose transport protocol due to its innovative characteristics. The initiation procedure of this protocol uses a 4-steps cookie exchange mechanism, which grants immunity to flood attacks. Multistreaming allows transmission of several data streams within the same communication, splitting the application data into multiple streams that have the property of independently sequenced delivery, so that message losses in any one stream will only initially affect delivery within that stream, and not delivery in other streams. This is achieved by making independent data transmission and data delivery. SCTP uses a Transmission Sequence Number (TSN) for data transmission and detection of message losses, and also a Stream ID/Stream Sequence Number pair, which is used to determine the sequence of delivery of received data. Therefore in reception, the endpoint can continue to deliver messages to the unaffected streams while buffering messages in the affected stream until retransmission occurs.

The SCTP heartbeat mechanism allows endpoints to know the availability of each other, thus preventing unnecessary sends. At regular intervals, a heartbeat packet is sent to the remote IP addresses, so the source knows if a remote IP address is active or inactive. There is a counter for each remote IP address that counts how many times a heartbeat packet does not reach that remote IP address. If the counter exceeds a maximum value then the remote IP address changes to inactive and the SCTP protocol uses one of the alternative addresses.

Finally, multihoming allows each endpoint to use several IP addresses associated with the same communication, thus a session can remain active even in the presence of network failures. One of the main advantages is that in a conventional single-homed session, the failure of a local LAN access can isolate the end system, but with multi-homing, redundant LANs can be used to reinforce the local access. Observe that multi-homing is not used for redundancy. One of the IP addresses is selected as primary, and it will be used as destination in a normal transmission. If the heartbeat mechanism (monitoring function) detects that a route is not longer available, then another IP address is used as destination.

Dynamic address reconfiguration [13] is a step forward. It is a new feature that allows SCTP to use IP addresses not previously declared for the current communication. This feature becomes especially interesting in mobile communications, where IP addresses change dynamically. However, security issues arise when trying to apply this scheme with IPSec. Indeed, the use of IPSec with dynamic address reconfiguration SCTP implies, so far, the renegotiation of the IPSec SA parameters, i.e. the old tunnel is no longer available and a new one should be set up.

2.4. SCTP-IPSec compatibility

Essentially, the inconsistency between SCTP and IPSec can be described as follows: SCTP multihoming is not directly supported by using one-way IPSec SA. The reason is clear. An IPSec SA is univocally identified by the tuple {SPI, destination address, security protocol}. That is, one destination address for each SA. Thereby, multihoming is not included by definition. First approach to overcome this situation is creating as many SA as IP address are allowed by the SCTP endpoints. Nevertheless, taking into account new features for SCTP such as dynamic address resolution, this is not a viable solution. It is not feasible making SA if IP addresses are not even known.

A solution to this problem is presented in [9]. It proposes modifying the structure of the SA so that it can manage more than one IP address. Then, a SA is able to store several individual IP addresses, hereby, giving support to SCTP multihoming. The SAD usually saves these addresses within a SA in the form of a linked list (Fig. 5). However, this approach will likely experience a delay in the processing time required to check the SA parameters. Certainly, if the endpoint uses a secondary IP address, the remote endpoint needs to search the corresponding linked list of the SA in the SAD to verify if the new IP address in use is included. In next section we present an easy way to enhance this behavior.

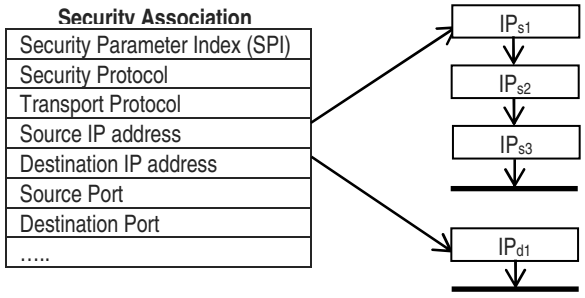

Figure 5. A Security Association with linked lists of IP addresses. Each IP address stores a poiner to the next IP address.

3. Improving IPSec-SCTP interaction

In this section we discuss our proposal to improve SCTP-IPSec interaction, and show an experimental implementation with satisfactory results.

3.1. Operation

To enhance SCTP-IPSec compatibility, we propose a simple further step, allowing each SA to store a complete range of IP addresses, i.e. a pool of IP addresses, even if the terminal initially only uses one of them.

The operation of this method is presented in Fig. 6. Assume we establish a secure SCTP connection, using IPSec, between two endpoints, A and B, located in different networks. A has two IP addresses: IP_{A1} 192.0.1.65/26, and IP_{A2} 192.0.1.66/26. B has also two IP addresses: IP_{B1} 192.0.2.65/26, and IP_{B2} 192.0.2.66/26. In order to establish a secure SCTP communication via IPSec, we need two SA (one for each direction) called SA_{AB} and SA_{BA}. Lets us compare the SA following the approach described in [9] or using our proposal (for instance with the ESP security protocol):

- From [9], the SA_{AB} is described by {11, {192.0.2.65, 192.0.2.66}, ESP}, and the SA_{BA} is described by {4, {192.0.1.65, 192.0.1.66}, ESP}. See Fig.6.a).
- With our proposal, the SA_{AB} is described by {11, 192.0.2.64/26, ESP}, and the SA_{BA} is described by {4, 192.0.1.64/26}, ESP}. See Fig.6.b).

The advantage of our approach is that both terminals could change their IP address as many times as needed (but always inside the specified range) within the same IPSec tunnel (keeping the same SA) without producing a security problem.

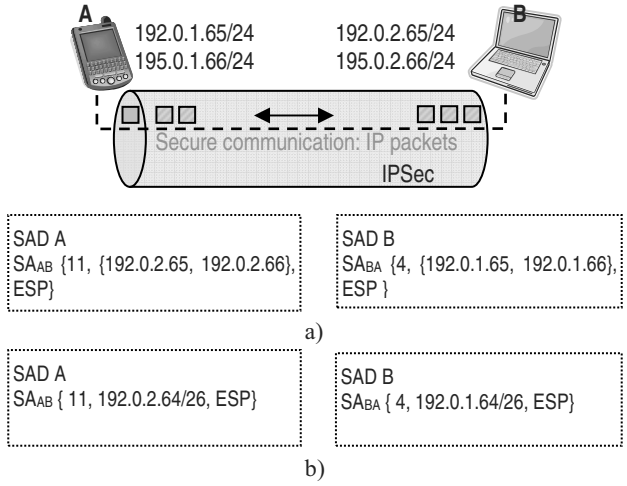

Figure 6. Example of SA. a) Using the approach from [9]; b) Using our proposal.

For instance, assume that there is a terminal A in network 192.0.1.64/26 communicating with two equipments, B1 and B2, in network 192.0.2.64/26 (see Fig. 7). Then, the SAD in A has two entries, one for transmission A→B1 and one for transmission A→B2. The SA_{AB1} is defined by {11, 192.0.2.64/26, ESP} and SA_{AB2} is defined by {12, 192.0.2.64/26, ESP}. The SAD in B1 has just one SA defined by {4, 192.0.1.64/26, ESP}; likewise, the SAD in B2 has just one SA defined by {4, 192.0.1.64/26, ESP}. Observe that SPI are unique locally, i.e., only within the same security association database. In this case, we suppose the SPI is repeated in B1 and B2, but they could be different. Assume an IP packet protected with the SA_{AB1} is sent (maliciously) to B2. This equipment is

going to find an entry in the SAD, since the SA_{B2A} has the same SPI (equal to 4) as SA_{B1A}, the same origin IP (192.0.1.64/26), and same security protocol (ESP). However, the security parameters (keys for authentication, keys for encryption, and maybe even algorithms to authenticate and encrypt) for SA_{B2A} are different from the security parameters for SA_{B1A}. Therefore, we do not incur a security threat by the fact of using a range of IP addresses instead of unique IP addresses to identify a security association.

In addition, suppose there are two WLAN whose coverage overlaps with UMTS coverage. Using the proposal from [9] (Fig. 8.a), it would be required to include in both, SA1 and SA2, all individual available IP addresses that users can employ for this communication. With our improvement (Fig. 8.b), if IP addresses used within UMTS cell1 were in the same range as those used in WLAN1 (not necessarily in the same network), users could move freely from WLAN1 to UMTS cell1, changing IP addresses if needed but using the same SA. The same applies if communication is established between a UMTS cell and an overlapped WLAN. In this last case, vertical handover with SCTP could be done faster in a secure way by using our compatible version of IPSec. Clearly, in any of the examples shown before, the processing time required to check if an IP address belongs to the SA in use is reduced with our approach.

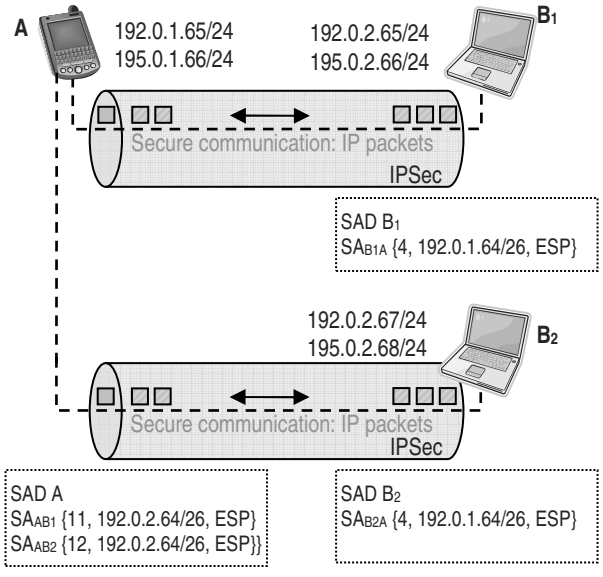

Figure 7. Example of two communications with same origin and different destination.

3.2. Implementation and experimental tests

In order to support our proposal, some straightforward changes should be done. Observe that to develop this IPSec implementation it is necessary to modify the common structure and processing of SA, and the IKE protocol behavior.

Regarding the first part, structure and processing of SA, our approach incorporates a very simple method, not used so far, to store feasible IP addresses within a SA: a pair [IP address, netmask] represented in Fig. 9. As well as decreasing processing time, the SAD size is also reduced. This is the system incorporated in our IPSec implementation, with a completely satisfactory result tested by experimentation.

Concerning IKE, all IP addresses that are likely to be used have to be known by the other communication endpoint. We propose that this exchange could be done during the negotiation of the SA for IPSec. Notice that this exchanged is secure thanks to the security association of ISAKMP. More specifically, our implementation includes these likely IP addresses in the Identity payload (Fig. 4) of the ISAKMP messages that are exchange in the second ISAKMP phase. Observe that addresses are specified as pairs [IP address, netmask].

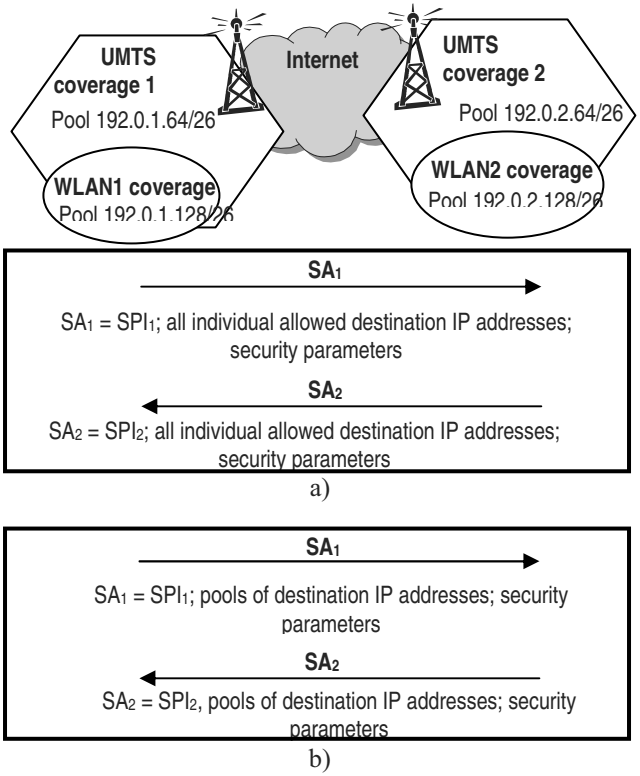

Figure 8. Example of the advantages of SCTP multihoming and IPSec.

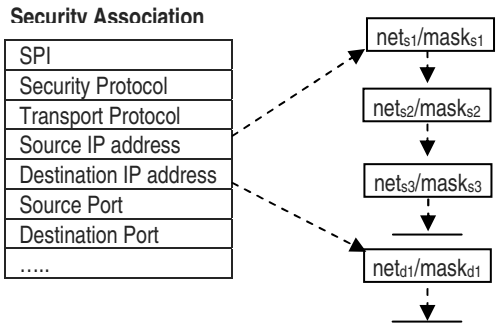

Figure 9. A Security Association with linked lists of pairs [network, mask]. To store a unique IP address, just set the proper netmask.

Our IPSec implementation, written in C code, was designed to operate in the user memory zone of a Linux terminal (SuSe 10). This IPSec implementation employs raw sockets to send messages, and the Libpcap library to receive messages.

Raw sockets are not linked to any transport port, so they can send raw data. With raw sockets, the programmer can create the IP header so that the Kernel only adds the network layer (e.g. Ethernet). Using raw sockets to receive data was not useful, because the IP header is automatically discarded. Moreover, all traffic would be received by the PC running our implementation regardless the destination inside the PC. For this reason, we employ Libpcap functions in reception, since they include filters that operate at Kernel level.

Tests were conducted in the topology depicted in Fig. 10. This topology comprises one laptop as a SCTP client with two network interfaces, and three PCs: a SCTP server, Router$_1$, and Router$_2$. The laptop and the PCs have Linux operating system. The SCTP client has two IP addresses available, 192.168.2.1 (wireless) as primary address and 192.168.3.1 (wired) as secondary address, therefore being able to use the multihoming capability. The SCTP client downloads several files from the SCTP server, using the same SCTP connection (this application was developed by the authors in a previous work).

In this scenario, we use IPSec to establish a secure tunnel between the SCTP client and Router$_2$. The tunnel is represented in Fig. 10 with a gray area. The SA from Router$_2$ to the SCTP client includes the pairs 192.168.2.0/ 255.255.255.0 and 192.168.3.0/ 255.255.255.0 as addresses to be protected with the same security parameters.

To verify the goodness of our IPSec implementation efficiently compatible with SCTP, we force a failure in the SCTP client's network adapter with IP address 192.168.2.1 (primary address). The SCTP client sets then its secondary IP address (192.168.3.1) as the active one. After this change, a regular IPSec implementation would disable the tunnel, and it would require a new negotiation of IPSec tunnel parameters before setting the tunnel again. With our IPSec implementation, the same IPSec tunnel is kept giving an uninterrupted security service to the SCTP connection. As an example, Fig. 11 shows a traffic capture made with Wireshark [14] during the tests, where we can observe that after changing the IP address, the SPI of the IPSec tunnel is the same. Several configurations for the IPSec tunnel were tested in this scenario (AH plus ESP with and without authentication, DES or 3DES, etc.), all reporting successful results.

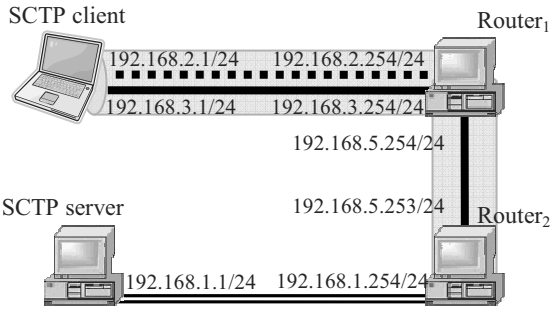

Figure 10. Topology for tests. The gray area represents the path protected by our IPSec implementation. Solid line is a wireline connection, and the dotted line is a wireless connection.

SPI	0x80300000
Security Protocol	ESP+authentication
Transport Protocol	SCTP
Source IP address	192.168.1.1/32
Destination IP address	192.168.2.0/24 192.168.3.0/24
Source Port	Source Port
Destination Port	Destination Port

a)

Time	Source	Destination	Protocol	Info
0.000000	192.168.1.1	192.168.2.1	ESP	ESP (SPI=0xe8030000)
0.003021	192.168.1.1	192.168.2.1	ESP	ESP (SPI=0xe8030000)
5.978497	D-Link_31:d		ARP	who has 192.168.3.1?
5.978531	AniCommu_01		ARP	192.168.3.1 is at 00
5.979489	192.168.1.1	192.168.3.1	ESP	ESP (SPI=0xe8030000)

b)

Figure 11. a) Data stored in a Security Association within our enhanced IPSec. b) Traffic capture during a network failure: the primary IP address becomes inactive. SCTP connection uses the secondary IP address. The IPSec tunnel is the same even with the change of IP addresses in use.

4. Conclusion and further work

In this work, we present an enhanced IPSec implementation able to efficiently work with multihoming SCTP. In order to do so, we have redesigned the SA used by IPSec so that they can store several IP addresses, either single or [network, mask] pairs. This latter option significantly improves the performance compared with the use of multiple single IP addresses. From a qualitative point of view, the time needed by the SAD to check if an alternative source or destination IP address belongs to the same SA is smaller if checking is done by sets of IP addresses rather than by individual ones. In addition, the IPSec tunnel can be kept if the user uses an IP address from a predefined pool of addresses, without the need of exactly specifying what the IP addresses in use are. Despite the conceptual simplicity of our approach, it has not been proposed before in related literature about IPSec or SCTP security. Currently, we are working towards a solution to allow IPSec and SCTP with dynamic address reconfiguration.

5. Acknowledgements

This research has been supported by project grant TEC2007-67966-01/TCM (CON-PARTE-1) and it is also developed in the framework of "Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, de la Fundación Séneca, Agencia de Ciencia y Tecnología de la RM (Plan Regional de Ciencia y Tecnología 2007/2010).

6. References

[1] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[2] C. Kaufman, “Internet Key Exchange (IKEv2) Protocol”, RFC 4306, December 2005.

[3] R. Stewart et al., "Stream Control Transmission Protocol", IETF RFC 2960, October 2000.

[4] S. Fu, M. Atiquzzaman, "SCTP: State of the Art in Research, Products, and Technical Challenges", *IEEE Communications Magazine*, April 2004.

[5] M. Li, Y. Fei, V.C.M. Leung, T. Randhawa, “A new method to support UMTS/WLAN vertical handover using SCTP”, *IEEE Wireless Communications*, Vol. 11, Issue 4, pp. 44-51, August 2004.

[6] E-C. Cha, H.-K. Choi, S.-J. Cho, “Evaluation of Security Protocols for the Session Initiation Protocol”, Proc. International Conference on Computer Communications and Networks ICCCN’07, pp. 611-616, August 2007.

[7] E. Unurkhaan, E. P. Rathgeb, A. Jungmaier, “Secure SCTP – A Versatile Secure Transport Protocol”, *Telecommunications Systems*, Vol. 27, No. 2-4, pp. 273-296, October 2004.

[8] C.-Z. Yang, W.-K. chang, I.-H. Huang, “CS-SCTP: A Collaborative Approach for Secure SCTP over Wireless Networks”, Proc. IEEE TENCON’07, pp. 1-4, November 2007.

[9] Bellovin et al., "On the Use of Stream Control Transmission Protocol (SCTP) with IPsec", RFC 3554, July 2003.

[10] H. Orman, “The OAKLEY Key Determination Protocol”, RFC 2412, November 1998.

[11] H. Krawczyk, “SKEME: A Versatile Secure Key Exchange Mechanism for Internet”, Proc. Symposium on Network and Distributed System Security SNDSS '96, pp. 1-14, 1996.

[12] D. Maughan, M. Schertler, M. Schneider, J. Turner, “Internet Security Association and Key Management Protocol (ISAKMP)”, RFC2408, Noviembre 1998.

[13] R. Stewart, Q. Xie, M. Tuexen, S. Maruyama, M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, September 2007.

[14] Wireshark. <www.wireshark.org>. April 2008.